Введение в математическую логику

Лекция 1

0. Что такое математическая логика

Логика — попытка упорядочить человеческие рассуждения — существует несколько тысяч лет.

В какой-то момент математики задали вопрос: «В чём, собственно, состоит математика, математическая деятельность?» Простой ответ заключается в том, что математики доказывают теоремы, то есть выясняют некоторые истины о реальном мире и/или «идеальном математическом мире». Попытка ответить на вопрос, что такое математическая теорема, математическая истина и что такое математическое доказательство, что значит выяснить, что некоторое математическое утверждение истинно или доказуемо, — это и есть исходная точка математической логики.

Второй исходной точкой наших рассмотрений является выяснение того, что значит, что математическая функция вычислима, то есть её можно вычислить с помощью некоторого алгоритма, формального правила, точно описанной процедуры.

У этих двух исходных постановок и в исследовании их есть много общего, они естественно объединяются под общим названием «математическая логика», где под математической логикой понимается прежде всего логика математических рассуждений и математических действий. Однако часто математическую логику делят на собственно математическую логику (у которой тоже есть своё деление) — науку о математическом описании математических рассуждений и их результатов, и на теорию алгоритмов — математическое описание того, что такое вычисление, алгоритмы, вычисляемые ими функции и так далее.

В большой степени проблематика математической логики возникла на рубеже XIX и XX века. Великий математик Давид Гильберт (23.01.1862 – 14.02.1943), один из величайших, а может быть, наряду с Андреем Николаевичем Колмогоровым, величайший математик XX столетия, на рубеже XIX и XX века поставил вопрос о том, что такое геометрическая теорема и геометрическое доказательство, и написал замечательную книжку «Основания геометрии» (1899).

Всем, кто проходил геометрию в школе, объясняли, что геометрия состоит в том, чтобы доказывать теоремы, исходя из заданных аксиом. Гильберт, начав изучать классическую геометрию Евклида, близкую к которой до сих пор проходят в школах во многих частях мира, понял, что геометрия Евклида не является полностью формализованной наукой, а в ней ещё есть много неформальных, математически точно не описанных способов рассуждений.

Гильберт, изучив этот вопрос, построил более точную, более ясную, более полную систему доказательств геометрии, аксиом и тех правил, по которым доказательства строятся. Но Гильберт задался вопросом и о том, как вообще устроена не только геометрия, но и вся математика, к тому времени существовавшая (а это очень существенная часть всей математики, которую сейчас изучают в университетах; одним из важнейших добавлений к ней как раз и явилась математическая логика).

На Втором Международном математическом конгрессе, который проходил в Париже в 1900 году, Гильберт сформулировал 23 проблемы, которые ему казались важнейшими проблемами математики того времени. Из этих 23 проблем несколько, по крайней мере четыре, имеют непосредственное отношение к тому, что сегодня называется математической логикой. Многие из этих 23-х проблем, бóльшая их часть, были решены, многие из них были решены российскими, советскими математиками, в частности и проблемы, относящиеся к математической логике и теории алгоритмов. В этом курсе мы познакомимся с некоторыми из проблем Гильберта и с тем, как они решались.

Важнейшая задача, которую видел перед собой Гильберт, — это полностью описать, что такое математика как формальная, математически ясная система рассуждений. До него этот вопрос не осознавался как столь важный, центральный для всей математики. Существенные результаты в решении данной задачи ещё до Гильберта были получены другими математиками, в частности, Готлобом Фреге (1848 – 1925). Большая степень ясности была достигнута в 30-е годы прошлого века, то есть 70 – 80 лет назад, существенные результаты были получены в 1960-е гг.

Идея Гильберта состояла в том, что математическая деятельность — это получение цепочек символов по математически определённым правилам. И результат этой деятельности, то есть математическое доказательство как объект — это система цепочек, полученных по определённым правилам. Ближе к концу концу курса мы в явном виде выпишем систему правил, лежащую сегодня в основе всей математики. Это — так называемая аксиоматическая теория множеств. Вы увидите, что эта система правил не очень большая, она занимает всего пару страниц. И этой системы достаточно для получения многих тысяч, миллионов страниц математических доказательств. Это весьма замечательный факт о реальной жизни математической науки.

Программа построения системы математических доказательств, кратко описанная выше, называется сегодня программой Гильберта. Ещё одна задача, которую Гильберт считал важной, это доказательство того, что в Системе математических доказательств нельзя получить противоречие. Мы уточним в дальнейшем, что это значит, но идея заключается в том, что какие-то вещи — правильные — мы можем доказать, получить в этой системе, а неверные, неправильные, — не можем. Такова была исходная мечта Гильберта, и были предложены некоторые подходы к тому, как убедиться в том, что математика непротиворечива, и что в ней ничего страшного доказать нельзя.

На самом же деле выяснилось, что эта надежда Гильберта неосуществима, и это было доказано тоже математическими средствами, наряду с другими очень важными утверждениями математической логики, Куртом Гёделем. Давид Гильберт был немецким математиком, он родился в городке Велау близ Кёнигсберга в Восточной Пруссии (теперь посёлок Знаменск Калининградской области) и почти всю свою жизнь прожил в Гёттингене, а упомянутые результаты о том, что что-то в математике невозможно получить, принадлежат другому математику, австрийскому, родившемуся в Австро-Венгрии, в чешском городе Брно, Курту Гёделю.

1. Алфавит, буква, слово, язык

Итак, математическая логика посвящена анализу результатов действий, которые выполняются по формально описанным математическим правилам. При этом, интересуясь доказательствами, мы рассматриваем системы правил, которые разрешают нам выполнять некоторые действия. Такие системы называются *исчислениями*, и мы постепенно будем уточнять наше понимание того, что это такое.

Рассуждения, доказательства, которые проводят математики, используют какой-то человеческий, естественный язык, можно считать, что различные математические символы относятся к естественному языку (русскому, английскому и т. д.), можно считать, что они к нему добавляются, для нас это не так существенно. В любом случае, терминология, которая используется в этой части математической логики, навеяна исследованиями естественных языков, лингвистикой. Самые первые термины, которые мы собираемся ввести, носят лингвистический характер, хотя они описывают чисто математические и очень простые понятия.

Первое понятие — это алфавит. Начиная строить какой-либо язык, мы начинаем с его алфавита.

Определение 1. *Алфавит* — это любое непустое множество. Его элементы мы будем называть *символами*, или *буквами.*

Определение 2. *Слово* в алфавите Σ — это любая конечная последовательность элементов алфавита Σ.

Пока мы не придаём никакого смысла, или семантики, понятию слова.

Определение 3. *Язык* в алфавите Σ — это любое множество слов в алфавите Σ.

С понятием «язык» мы, опять-таки, не связываем никаких содержательных соображений. Язык — это просто любое множество слов: конечное, пустое, бесконечное, — любые множества годятся.

Пример 1. $baaa$ — слово в алфавите $\Sigma = \{a, b, c\}$.

Пример 2. 386 — слово в алфавите десятичных цифр $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Возможны и другие алфавиты. Например, мы можем добавлять в алфавит знаки препинания, и т. д. Как можно заметить, в слове мы пишем символы подряд, без запятых, точек, пробелов или каких-то иных разделителей.

Среди слов удобно и полезно иметь ещё так называемое пустое слово, это слово, в котором вообще нет букв. Это слово является словом в любом алфавите.

Определение 4. Слово, не содержащее ни одного символа, называется *пустым словом* и обозначается Λ.

Если взять любое слово в каком-либо алфавите и начать вычёркивать по одной все буквы, то рано или поздно мы получим пустое слово.

Определение 5. *Длиной* слова w называется число символов в слове w, то есть число символов в конечной последовательности. Длина слова w обозначается $|w|$, подобно тому, как в алгебре обозначается модуль числа.

Примеры: $|baaa| = 4$; если $w = 386$, то $|w| = 3$; $|\Lambda| = 0$.

Следующее простое и наглядное понятие — это приписывание одного слова к другому.

Определение 6. Если x и y — слова, то слово xy — это результат *приписывания* к слову x слова y.

Чуть более тонким является понятие того, что одно слово входит в другое.

Определение 7. Говорят, что слово v *входит* в слово w, если слово w можно представить в виде $w = uvs$, где u и s — некоторые слова.

Одно и то же слово v может несколько раз входить в слово w, и нам хотелось бы выделять разные вхождения и говорить, например, о первом вхождении, о втором, третьем, …, последнем. Для этого придумали способ, выглядящий несколько искусственно или вычурно для обычного человека, — по краям того слова, которое входит, ставить звёздочки, и слово

$$u * v * s$$

называть вхождением. Значит, вхождение — это тоже слово, но в более широком алфавите, чем первоначальный, так как мы считаем, что звёздочка в первоначальный алфавит не входит.

Определение 8. Пусть w — слово в алфавите, не содержащем звёздочку, и пусть $w = uvs$. Тогда слово $u * v * s$ называется *вхождением* слова v в слово w.

По аналогии с обычной алгеброй мы определяем n-ю степень слова x.

Определение 9. Пусть x — слово, а n — натуральное число. Тогда n-й *степенью* слова x называется результат последовательного приписывания друг к другу n копий слова x, и получающееся слово обозначается через x^n. Полагаем, что x^0 — это пустое слово.

Примеры: $ba^3 = baaa$; $(ba)^3 = bababa$; $(ba)^0 = \Lambda$.

2. Грамматика

Мы подошли к первому содержательному понятию, несложному, но важному и довольно глубокому — понятию грамматики. Как уже говорилось, многие вещи здесь связаны с обычными человеческими языками, и понятие грамматики в том виде, в котором оно будет приведено, возникло при попытке описать не математический язык, а естественный человеческий язык, а именно, английский. Здесь нужно назвать имя Ноама Хомского (7.12.1928 –). Это — один из самых знаменитых лингвистов XX в., американец, живущий и работающий в Массачусетском технологическом институте в Соединённых Штатах. В конце 50-х годов, он придумал понятие грамматики, которое мы сейчас рассмотрим.

В грамматике имеются алфавиты. Один алфавит называется основным, а другой называется вспомогательным.

В грамматике Хомского основной алфавит состоял не из букв, а из слов английского языка, а вспомогательный алфавит состоял из таких лингвистических терминов, как группа подлежащего или дополнение.

Вспомогательный алфавит не пересекается с основным алфавитом, а объединение этих двух алфавитов называется алфавитом грамматики.

Центральная часть понятия грамматики — это правило. Мы уже говорили о том, что математическое рассуждение происходит по некоторым правилам. Сейчас мы впервые встречаемся с понятием правила, описанным математически, и оказывается, что понятие правила очень простое. Правило — это два слова, между которыми стоит стрелочка. Сама стрелочка не есть символ основного или вспомогательного алфавитов (она, как и звёздочка, которую мы вводили для обозначения вхождений, — некоторый новый символ). И наконец, один из символов вспомогательного алфавита мы называем начальным и обозначаем его S.

Если определить формально, что такое грамматика, то это четвёрка: два алфавита, система правил и начальный символ.

Определение 10. *Грамматика* Γ — это набор $\langle \Sigma, \Omega, \Pi, S \rangle$, где
Σ — *основной алфавит* грамматики Γ,
Ω — *вспомогательный алфавит* грамматики Γ,
$\Sigma \cap \Omega = \varnothing$,
$\Sigma \cup \Omega = \Delta$ — *алфавит* грамматики Γ,
Π — конечное множество слов вида $u \to v$, называемых *правилами* грамматики Γ, в которых u и v — слова в алфавите Δ, а $\to \notin \Delta$,
S — *начальный символ* грамматики Γ, $S \in \Omega$.

Следующее, тоже довольно простое, определение, но центральное для очень большой части математической логики и Computer Science — это применение правила к слову.

Определение 11. *Применить* правило $u \to v$ к слову $w = tus$ в алфавите Δ — это значит из слова tus получить слово tvs.

Это довольно понятная вещь, с которой мы постоянно сталкиваемся в алгебре (в том числе — школьной), когда просто кусочек выражения мы заменяем на другое выражение. Мы ищем вхождение слова u в слове w и применяем к нему правило, то есть заменяем u на v. Обратим внимание на то, что в слове u может быть несколько букв, как и в слове v, что слово u может быть и пустым, как и слово v. И ещё важная и интересная вещь, что слово u может иметь несколько вхождений в слово w. И когда мы применяем правило, мы можем применить его по-разному, то есть в разных местах заменить u на v. Любая такая замена (но одна замена!) называется применением правила.

На основе понятия правила и понятия применения правила к слову можно дать определение выводимого слова:

Определение 12.
1. Начальный символ грамматики — *выводимое* слово.
2. Результат применения правила к выводимому слову — *выводимое* слово.

Понятие выводимого слова можно определить и несколько иначе, но эквивалентно, привлекая понятие вывода:

Определение 13. *Выводом* в грамматике Γ называется конечная последовательность слов в алфавите Δ, в которой первое слово есть S, а каждое следующее слово получается из предыдущего применением некоторого правила из Π.

Определение 14. Слово называется *выводимым* в грамматике, если для него существует вывод, то есть такой вывод, в котором это слово — последнее.

Определение 15. Слово называется *породимым* в грамматике, если оно выводимо и не содержит символов из Ω (вспомогательного алфавита).

То есть, вспомогательные символы на то и вспомогательные, что нужны только по ходу вывода, а в конце должны исчезнуть.

И последнее определение в этой серии:

Определение 16. *Языком, порождаемым грамматикой,* называется язык, состоящий из всех слов, породимых в грамматике.

Рассмотрим примеры грамматик.

Пример 1. $\Gamma = \langle \Sigma, \Omega, \Pi, S \rangle$,
основной алфавит $\Sigma = \{0, 1\}$,
вспомогательный алфавит $\Omega = \{S\}$,
множество правил вывода $\Pi = \{S \to S0, \quad S \to S1, \quad S \to \Lambda\}$.

В этом примере основной алфавит состоит из двух символов: 0 и 1. Значит, наши породимые слова будут в этом алфавите. Одно правило состоит в том, что если где-то имеется символ S, то мы можем поставить после него 0. Другое правило позволяет после S ставить 1. И есть правило, позволяющее S просто стереть, то есть заменить на пустое слово.

Легко понять, что данная грамматика позволяет порождать любые цепочки из нулей и единиц. Вот пример вывода: $S \to S0 \to S10 \to S010 \to S0010 \to 0010$.

Задача 1. Построить грамматику, порождающую все десятичные числа. Пример числа: -3.141592.

При решении этой задачи нужно учесть, что в записи десятичного числа может иметься десятичная точка (или, если хотите, запятая), а может не иметься. В начале может стоять минус, а может и не стоять. Ноль в начале десятичного числа может быть только в особых случаях, если число равно нулю или если сразу после нуля стоит десятичная точка.

Нужно обдумать, как будет идти процесс порождения, чтобы учесть все эти обстоятельства. В нашем простейшем случае, примере 1, нужен был только один вспомогательный символ S, он же был и начальным символом. А в этой задаче понадобятся несколько вспомгательных символов.

Пример 2. $\Gamma = \langle \Sigma, \Omega, \Pi, S \rangle$,
основной алфавит $\Sigma = \{a\}$,
вспомогательный алфавит $\Omega = \{S, B, M, E\}$,
множество правил вывода
$\Pi = \{S \to BaE, \quad B \to BM, \quad Ma \to aaM, \quad ME \to E, \quad B \to \Lambda, \quad E \to \Lambda\}$.

В этом примере начать вывод можно только с применения первого правила, так как только в первом правиле слева стоит S (и нет правил с пустой левой частью). После применения первого правила получится слово BaE. Буква B может отщеплять от себя M (второе правило). Буква M может двигаться вправо, перепрыгивая через a, но удваивая при этом число букв a, то есть каждый раз заменяя a на aa (третье правило). Когда M доходит до E, то оно может исчезнуть (четвёртое правило), и это единственный способ для вспомогательного символа M исчезнуть — добежать до буквы E в конце. Фокус здесь в том, что мы рассматриваем выводы только таких слов, в которых нет вспомогательных символов, а для того, чтобы у нас не осталось вспомогательных символов, M должно добежать до конца. В процессе вывода у нас будут возникать слова с различным количеством символов a, но в самом конце, когда не останется вспомогательных символов, количество символов a должно быть степенью двойки.

Это пример грамматики, порождающей язык, состоящий из слов длины 2^n для любого n.

Упражнение. Проделать аккуратное рассуждение для примера 2.

Задача 2. Построить грамматику, порождающую все слова, состоящие из одинакового количества букв a и b.

И вот задача, которая дальше будет нас интересовать — это задача о невозможности:

Задача 3. Построить язык, который породить нельзя.

У нас ещё будут возникать ситуации невозможности. Это очень важно. В начале лекции была упомянута программа Гильберта, состоящая в том, чтобы всю математику описать, упорядочить и объяснить, что она хорошо и надёжно устроена. Но это невозможно, и это доказал Гёдель.

С утверждениями Гёделя имеет непосредственную связь и попытка построить язык, который нельзя породить. На самом деле, существование таких языков — вещь довольно очевидная, а вот явное построение — более сложно.

3. Термы. Индуктивные определения. Синтаксис и семантика

3.1. Что такое синтаксис и семантика. Имена

Если продолжать грамматические параллели, то вопросы о том, как устроены слова и предложения, обычно называются синтаксисом — слово, которое известно из школы. А придание текстам, словам и предложениям смысла называется и в лингвистике, и в обычном словоупотреблении, и в математике — семантикой. Формальное определения здесь не нужно — это общечеловеческий, общематематический язык, не требующий формального определения, а в частных случаях мы будем ясно математически задавать и синтаксис, и семантику.

Простейшим понятием семантики является понятие имени. Важное различение и в математике, и, до некоторой степени, в жизни, и в программировании — это то, что имеется *имя* объекта, и имеется значение имени — сам *объект*.

Некоторые имена в математике имеют фиксированное значение, которое, впрочем, иногда меняется.

Например, обычно значение имени π — это действительное число: отношение длины окружности к её диаметру, но иногда это функция, дающая число простых чисел в натуральном ряду, не превосходящих аргумента функции. Значением имени «+» обычно считается сложение натуральных, действительных, комплексных чисел и т. д., но иногда бывает и сложение векторов, а иногда ещё более тонкое и экзотическое употребление.

Есть некоторые способы говорить в нашем языке об именах, значениями которых являются слова. Значением слова «слон» в русском языке является животное, такое крупное, серое, а если мы хотим говорить о слове «слон», то мы ставим слово «слон» в кавычки.

И конечно, важную роль в математике играют имена чисел. Можно считать, что значением имени «23» является число 23, существующее в мире математических объектов, но можно считать, что значением имени «23» является оно само. А можно считать, что натуральные числа — это просто цепочки из десятичных цифр, не начинающиеся с нуля.

Бывают сложные имена. Например, « $2+4$ » — сложное имя, значением которого является число 6. Сейчас мы как раз и займёмся тем, как выявлять значение сложного имени, если мы знаем значение составляющих его простых имён. И ещё одно общее понятие — это *интерпретация.*

Интерпретация — это задание значений, или вычисление, нахождение значений имён. Часто в математической логике и вообще в математике фиксируется некоторая область интерпретации, область, в которой принимают значение имена предметов, или объектов.

В математике часто используется понятие функции. Для функций тоже имеются имена. Например, имя « $\sin$ » — это имя функции, которая имеет знакомый всем периодический график. Если фиксирована область интерпретации, то именам функций в этой интерпретации соответствуют функции на этой области.

3.2. Термы (синтаксис термов, индуктивное определение термов)

Мы подошли к следующему понятию — понятию терма. Мы начнём с синтаксиса этого понятия.

Мы рассматриваем алфавит имен предметов, или объектов

$$\mathsf{Ob} = \{a_0,\, a_1,\, a_2, \ldots\}.$$

Это множество символов, которые потом будут именами объектов в области интерпретации. Но пока мы их рассматриваем без всякой интерпретации, пока это просто термин — «имена предметов».

Ещё мы рассматриваем алфавит имен функций

$$\mathsf{Fun} = \{f_0,\, f_1,\, f_2, \ldots\},$$

причём мы считаем, что для каждой функции известно число, которое математики называют арностью, — число её аргументов. Иногда сами предметы тоже можно считать функциями с нулём аргументов. (Как, скажем, число 3 можно считать постоянной функцией без аргументов.) Но мы, скорее всего, не будем пользоваться функциональными символами нулевой арности.

И ещё нам нужен алфавит предметных переменных:

$$\mathsf{Var} = \{x_0,\, x_1,\, x_2, \ldots\}.$$

Понятие «переменная» тоже знакомо нам из математики, здесь ничего особенного не имеется в виду, кроме того, что это просто некоторые символы, тоже некоторые буквы, из специально заведённого нами алфавита.

И сейчас мы определим, что такое терм.

Определение 17.
1. Имя предмета, то есть любое a_i, — это *терм.*
2. Переменная, то есть любое x_i, — это *терм.*
3. Если f — имя n-арной функции, и $t_0,\ t_1,\ \ldots,\ t_{n-1}$ — термы, то $f(t_0,\ t_1,\ \ldots,\ t_{n-1})$ — *терм.*

Это — так называемое индуктивное определение. Индуктивные определения часто встречаются и в математике, и в программировании, а для математической логики — это одно из центральных понятий. Мы некоторые простейшие объекты объявляем чем-то, в данном случае термами, и указываем некоторый способ конструирования из простейших объектов более сложных.

Если мы вернёмся к определению грамматики, то увидим, что оно тоже имело характер индуктивного определения: если нечто уже выводимо, то мы можем в этом выводимом заменить кусочек на что-то другое и снова получим выводимое.

Часто к индуктивному определению добавляют ещё один пункт, так называемое ограничительное утверждение. В случае нашего определения терма это был бы четвёртый пункт, в котором говорилось бы, что то, что получается в соответствии с предыдущими пунктами — это термы, а всё остальное — не термы. Или же примером такого ограничительного утверждения могло бы быть следующее: «Слово может быть термом только по одной из трёх указанных причин». Или: «Множество термов — это наименьшее множество, удовлетворяющее трём указанным условиям». Мы для краткости не стали вставлять в определение терма таких подразумевающихся ограничительных утверждений.

Пример терма: $f_4(f_1(a_1,\ f_2(a_5, f_4(x_2)),\ x_2))$.

Выписанное здесь слово является примером терма при следующих предположениях об арностях входящих в него функциональных символов (и соответствующих им функций):
— арность f_4 равна 1, такие функции называются *унарными*,
— арность f_2 равна 2, такие функции называются *бинарными*,
— арность f_1 равна 3, такие функции называются *тернарными*.

Этот терм строится за 4 шага:
1) так как x_2 — терм, а арность f_4 равна 1, то $f_4(x_2)$ — терм,
2) так как a_5 и $f_4(x_2)$ — термы, а арность f_2 равна 2, то $f_2(a_5, f_4(x_2))$ — терм,
3) так как a_1, $f_2(a_5, f_4(x_2))$ и x_2 — термы, а арность f_1 равна 3, то $f_1(a_1,\ f_2(a_5, f_4(x_2)),\ x_2)$ — терм,
4) так как $f_1(a_1,\ f_2(a_5, f_4(x_2)),\ x_2)$ — терм, а арность f_4 равна 1, то $f_4(f_1(a_1,\ f_2(a_5, f_4(x_2)),\ x_2))$ — терм.

3.3. Семантика термов, индуктивное определение значения терма

Интуитивно семантика термов совершенно понятна — это просто функции, композиции функций. Но когда мы начинаем заниматься уточнением того, в чём состоит смысл того или иного термина, требуется некоторая аккуратность.

Естественно, что сначала нужно задать *область интерпретации*. Скажем, если у нас есть функция сложения, то, как уже говорилось, это может быть сложением чисел, а может быть сложением векторов и т. д.

Дальше мы должны фиксировать значения в нашей интерпретации для каждого имени предмета. Значения мы берём из области интерпретации.

И точно также естественно, что для каждого имени функции мы просто фиксируем некоторую функцию на области интерпретации.

Что такое функция — это известно. Функция — это подмножество прямого произведения. Скажем, одноместная, унарная функция действительной переменной — это подмножество двумерной действительной плоскости, множество пар.

В математической логике обычно рассматриваются всюду определённые функции. А в теории алгоритмов, наоборот, важную роль играют не всюду определенные функции. Это различие будет где нужно подчёркиваться.

Определение 18. Пусть задан алфавит имён предметов Ob, и пусть задан алфавит имён функций Fun (и про каждое имя функции известна его арность). *Интерпретацией* называется пара $\langle M, \mathsf{Зн}\rangle$, где

M — непустое множество, называемое *областью*, или *носителем*, интерпретации,

$\mathsf{Зн}$ — функция, определённая на объединении $\mathsf{Ob} \cup \mathsf{Fun}$, которая каждому имени предмета ставит в соответствие его значение — некоторый элемент из M, а каждому имени функции арности n ставит в соответствие значение этого имени — некоторую всюду определённую функцию той же арности на M, (то есть унарную функцию из M^n в M, то есть некоторое подмножество прямого произведения M^{n+1}).

После того, как дано формальное определение интерпретации, начинает проясняться, какая функция соответствует терму. Но нам ещё нужно позаботиться о том, на каких значениях аргументов наша функция действует. Давайте зафиксируем бесконечную последовательность α элементов из M, носителя нашей интерпретации:

$$\alpha = \alpha_0, \, \alpha_1, \, \alpha_2, \ldots \, .$$

И теперь объясним, как, зафиксировав последовательность α, нам искать значения всех термов на этой последовательности. Определение будет дано индукцией по построению терма.

Определение 19. Пусть задана интерпретация $\langle M, \mathsf{Зн}\rangle$ и последовательность α. Тогда

1) *значение терма* a_i *на последовательности* α считается равным $\mathsf{Зн}(a_i)$,

2) *значение терма* x_i *на последовательности* α считается равным α_i,

3) *значение терма* $f_i(t_0, \, t_1, \, \ldots, \, t_{n-1})$ *на последовательности* α считается равным результату применения $\mathsf{Зн}(f_i)$ к значениям термов, стоящих в скобках.

Обратим внимание на то, что в этом определении три пункта. В определении терма тоже было три пункта: имя предмета является термом, переменная является термом, и имеется возможность сконструировать терм из n термов и функционального символа арности n. В точности по этим трём пунктам мы сейчас и идём, определяя значение терма. Такая конструкция называется индукцией по построению.

Итак, всякий терм отображает бесконечную последовательность α элементов множества M в один элемент множества M. Возникает функция из множества всех последовательностей в M. Множество всех бесконечных последовательностей элементов из M обозначим через M^ω, где $\omega = \{0, 1, 2, \dots\}$. Тогда терм задаёт функцию

$$M^\omega \to M.$$

Поскольку в терм входит только конечное число переменных, то от значений всех тех переменных, которые в терм не входят, функция, конечно, не зависит. Бесконечно много аргументов у этой функции — фиктивные. Но нам удобно считать, что терм задаёт функцию бесконечного числа аргументов.

Зададим теперь несколько странный, на первый взгляд, вопрос: однозначно ли определена функция, которую задаёт терм? Выражаясь иначе, можем ли мы всякий терм проанализировать однозначно? Строить термы мы умеем, но если нам дан терм, то можем ли мы однозначно восстановить процесс построения данного нам терма? Здесь есть некоторая трудность.

Мы подошли к принципиальному для математической логики вопросу. Этот вопрос возникает и в естественных языках. Имеется много примеров естественно-языковых фраз, в которых именно такой анализ неоднозначен. Часто мы с этим сталкиваемся и просто в реальном человеческом общении, но знаменитый литературный пример — это:

Казнить нельзя помиловать.

Должны ли мы понимать эту фразу как «Казнить нельзя и нужно помиловать», или «Казнить, а помиловать нельзя»? За проявлениями такого рода эффектов неоднозначности нужно очень тщательно следить в языках математики.

Оказывается, что термы однозначно анализируются, то есть в терме можно выделить единственным образом имя функции и единственным образом термы, которые стоят в качестве аргументов этой функции. Важным свойством терма является то, что число открывающих скобок в нём равно числу закрывающих. И это — ключевое свойство в доказательстве утверждения об однозначности анализа терма.

Упражнение 1. Доказать, что в терме число открывающих скобок равно числу закрывающих скобок.

Упражнение 2. Доказать утверждение об однозначности анализа терма.

Это очень простые, элементарные, упражнения, но важные с точки зрения понимания того, как устроены методы математической логики.

4. Общее понятие исчисления. Тезис Гильберта

Теперь уделим внимание общему понятию исчисления.

Эта общая идея не формализуема, так же, как, скажем, понятие множества. Мы просто привыкаем к тому, что такое множество, начинаем работать с множествами, но формального определения дать нельзя, а можно дать только некоторые пояснения. И способом избежать такой нечёткости является как раз введение аксиоматической теории множеств, о которой уже говорилось, которая была построена, в частности, Гильбертом.

У нас уже были примеры исчислений:

1) Было очень простое определение слова, породимого в грамматике, и оно относилось к простейшим математическим объектам — словам. Мы говорили про языки, порождаемые грамматиками.

2) У нас была конструкция индуктивного определения термов — исчисление термов. Можно рассматривать множества термов (с конечными алфавитами), порождаемые индуктивными определениями.

3) Говорилось об аксиоматической геометрии, но можно рассматривать и другие исчисления, возникающие в математике.

4) Можно строить исчисления, связанные с языками естественными. Строятся формальные грамматики, описывающие фрагменты естественного языка.

5) Много разных исчислений возникает в компьютерных системах.

И возникает общее интуитивное представление о том, что такое исчисление. Об этом уже упоминалось: это возможность применить некоторые правила для получения некоторых объектов, математических объектов.

Правила у нас включали в себя некоторое действие, но, как говорят математики, «не детерминированное», то есть не дающее однозначный результат (поскольку, например, заменить одно слово на другое в данном слове можно по-разному). В таких случаях удобнее говорить не о «недетерминированном действии», а об отношении. Получается, что исчисление — это два алгоритма:

1. Алгоритм допустимости — применяется к любому конечному множеству слов и ещё одному слову и дает ответ: Да или Нет.

2. Алгоритм завершения — применяется к одному слову и дает ответ: Да или Нет.

Определение выводимого в исчислении слова: слово x выводимо, если для некоторого множества слов Y, все элементы которого выводимы, алгоритм допустимости в применении к Y и x дает ответ Да. Слово z породимо, если оно выводимо и алгоритм завершения в применении к этому слову дает ответ Да.

Если мы говорим об общем, интуитивном понятии исчисления, то у нас возникает и интуитивное понятие о том, что такое *породимое* множество. (Слово «породимое» отсутствует в словаре русского языка, но используется в нашем курсе.) Исчисление позволяет считать некоторые слова *порождёнными*, и возникают языки *породимые*.

И оказывается, что всякое множество, породимое с помощью какого-то исчисления, может быть порождено некоторой грамматикой. Это утверждение — не математическая теорема. В это утверждение можно просто верить, это такой же естественно-научный факт, как закон всемирного тяготения, который не является математической теоремой, его можно вывести из других законов, но тогда возникнет вопрос, откуда мы взяли те законы, а вообще говоря, — это экспериментальный факт, многократно подтверждённый.

Утверждение, написанное ниже жирным шрифтом, тоже носит характер естественно-научного утверждения:

Всякое породимое множество порождается некоторой грамматикой.
(«Тезис Гильберта» — условно, в курсе)

Для всех тех исчислений, с которыми мы сталкиваемся, оказывается возможным построить грамматику, то есть исчисление очень просто описываемого вида, которое даёт в точности те же самые результаты.

Например, можно построить грамматику, порождающую все термы, если мы зафиксируем конечное множество предметных имён и конечное множество имён для функций.

Упражнение. Построить грамматику, порождающую термы.

Это не очень сложное упражнение.

Но замечательно, что и для всяких других, сколь угодно сложных исчислений, придуманных в математике или в изучении естественных языков, если эти исчисления формальны, формально описаны, формально заданы, то оказывается возможным для каждого такого исчисления построить грамматику, которая порождает в точности такой же язык, как и это исчисление.

Это утверждение можно называть тезисом Гильберта, но это название не является общепринятым. Оно использовалось и в других текстах по математической логике, но его найти не так легко в литературе или в интернете. Скорее, мы его приводим как параллель с другим утверждением, намного более известным, которое называется тезисом Чёрча, и о котором мы ещё будем говорить.